

WE CLAIM:

1 1. A method for enabling location independent and location
2 transparent interaction between a program and a user, the program
3 having been launched at a first location and having a program state
4 data structure for storing at least the program state, the method
5 comprising the steps of:

6 initiating a program status request by the user;
7 determining the current location of the program;
8 checking the program state to ascertain program status; and
9 interacting with said program based upon said program status.

1 2. The method of Claim 1 wherein said interacting with said
2 program comprises:

3 retrieving, from the program, output contents to display to
4 the user; and
5 displaying the output contents to said user.

1 3. The method of Claim 1 wherein said interacting with said
2 program comprises:

3 requesting input variables from said user;
4 sending any received input values to the current location; and
5 incorporating the received input values into said program
6 state data structure.

1 4. The method of Claim 1 wherein the program is a mobile
2 agent.

1 5. The method of Claim 1 wherein the program is a mobile
2 script.

1 6. The method of Claim 1 where the user is a mobile user.

1 7. The method of Claim 2 further comprising the step of
2 maintaining an output buffer and wherein said retrieving comprises
3 the step of retrieving the output contents from said output buffer.

1 8. The method of Claim 1 wherein the initiating step
2 comprises the steps of:

3 initiating the status request at a client machine; and
4 forwarding the status request to the first location at which
5 said program was launched.

1 9. The method of Claim 8 wherein said program comprises a
2 mobile program which executes a portion of its code at each of a
3 plurality of execution servers and wherein the determining step
4 comprises the steps of:

5 transmitting the status request to each execution server at
6 which the program has executed a portion of its code; and
7 determining, at each execution server, whether the program is
8 currently running locally.

1 10. The method of Claim 9 wherein each of said plurality of
2 execution servers maintains routing information for said program

3 and wherein said determining further comprises the step, if said
4 program is not currently running locally, of consulting said
5 routing information to ascertain at least one successive execution
6 server to which the program has been routed.

1 11. A method for enabling a user to provide input values to
2 a running program before the program needs the input values,
3 comprising the steps of:

4 maintaining a bag buffer of variable/value pairs in the
5 program;

6 receiving a communication, including input values, from the
7 user; and

8 temporarily storing said input values in said bag buffer.

1 12. The method of Claim 11 wherein said program subsequently
2 searches through contents of the bag buffer to locate needed input
3 values before requesting input from said user.

1 13. The method of Claim 2 further comprising the step of
2 maintaining a bag buffer in the program and wherein the retrieving
3 step comprises the steps of:

4 searching, in the bag buffer, for input values associated with
5 the input variables;

6 updating, if found, the input variables with the input values;

7 disposing, in an input buffer, the input variables, if not
8 found; and

9 optionally notifying the user via electronic means if no
10 suitable values are found in the bag buffer.

1 14. The method of Claim 13 wherein the electronic means is a
2 pager.

1 15. The method of Claim 13 wherein the electronic means is a
2 beeper.

1 16. The method of Claim 13 wherein the electronic means is
2 electronic mail.

1 17. The method of Claim 13 wherein the electronic means is a
2 smart telephone.

1 18. A computer program data structure comprising;
2 an output buffer for storing output values to be displayed to
3 a user;
4 an input buffer for storing values for which user input of
5 variables is required; and
6 a program state buffer for storing at least the present state
7 of said program.

1 19. The data structure of Claim 18 further comprising a bag
2 buffer for storing input variables.

1 20. The data structure of Claim 19 wherein the bag buffer is
2 a array data structure.

1 21. The data structure of Claim 19 wherein the bag buffer is
2 a hash table data structure.

1 22. The data structure of Claim 19 wherein the bag buffer is
2 a tuple space data structure.

1 23. An execution shell for a mobile program comprising:
2 a routing component for maintaining routing information
3 regarding said mobile program;
4 a processor component for processing user status requests
5 related to said program; and
6 an execution component for executing at least part of said
7 program.

1 24. The execution shell of Claim 23 further comprising a data
2 handling component for receiving user input and storing same in at
3 least one data structure for said program.